

## REMARKS

### SUMMARY

Reconsideration of the application is respectfully requested.

Claims 1-52 are in the application.

Applicants appreciatively acknowledge the Examiner's consideration and acceptance of the amendments made in Applicants' previous response. Applicants also appreciatively acknowledge the Examiner's withdrawal of the objection to Figure 10 and the Examiner's withdrawal of the rejection of claims 16, 17, and 22 under 35 U.S.C. § 101.

Applicants further appreciatively acknowledge the Examiner's consideration of the Applicants' previous arguments in "Response to Arguments," item 4 on page 3 of the above-identified final Office Action.

### DOUBLE-PATENTING

In "Double Patenting," item 14 on page 6 of the above-identified final Office Action, claims 1-52 have been *provisionally* rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 3-22, 24-30, 22, 35-38, 41-63, 66-75, 77-80, 83 and 84 of copending Application No. 10/082,807 (hereinafter '807) which was filed on the same day as the instant application and is owned by the same entity. Applicants respectfully traverse as '807 is explicitly drawn towards an annotation based development platform for **asynchronous** web services and not "stateful" web services as indicated in the instant application. More specifically, exemplary distinctions include that "stateful" web services would include a series of related web service requests, while "asynchronous" web services require coordination as they do not return immediate results.

When a double patenting rejection is appropriate, it must be based either on statutory grounds or nonstatutory grounds. The ground of rejection employed depends upon the relationship of the inventions being claimed. In the instant case, the above-identified Office Action indicates that a provisional nonstatutory obviousness-type double patenting rejection is being asserted. '807 is improperly identified as possessing conflicting claims, which are

not identical, but are allegedly not patentably distinct from the claims of the instant application.

Obviousness-type double patenting should only reject application claims when the claimed subject matter is not patentably distinct from the subject matter claimed in a commonly owned patent when the issuance of a second patent would provide unjustified extension of the term of the right to exclude granted by a patent. See *Eli Lilly & Co. v. Barr Labs., Inc.*, 251 F.3d 955, 58 USPQ2d 1865 (Fed. Cir. 2001); *Ex parte Davis*, 56 USPQ2d 1434, 1435-36 (Bd. Pat. App. & Inter. 2000). In the instant case, the claims of the ‘807 are drawn to “an asynchronous web service” as recited in claim 1 of ‘807 and not a “stateful web service” as recited in claim 1 of the instant application. Similar language distinguishing the two applications persists throughout. Withdrawal of the provisional rejection is respectfully requested.

Should the Examiner determine that the distinction between stateful web services and asynchronous web services are “not patentably distinct” as asserted by the Applicants, the Applicants will, upon issuance of either ‘807 or the instant application, submit the necessary Terminal Disclaimer for the remaining application. Thus, there will be no double patenting.

In “Double Patenting” item 15 on page 7 of the above-identified Office Action, claims 1-4, 10-12, 15-17, 22-24, 26, 31, 32, 34, 36, 38, 39 41, 44-46 and 48 have been *provisionally* rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1-8, 19-23, 26, 27, 31-36, 38, 39, 43, and 44 of copending Application No. 10/784,492 (hereinafter ‘492) which was filed after the instant application.

Like the previous item discussed above, Applicants respectfully traverse the provisional obviousness-type patenting rejection. Although, portions of the instant application may dominate the ‘492 application and vice versa, this domination in and of itself does not mandate a double patenting rejection. ‘492 is specifically directed towards creating network-based software services using source code annotations. More specifically, ‘492 adds Figures 11-15 and claims “an enhanced compiler capable of analyzing the annotated source code, recognizing numerous types of meta-data annotationism and generating a

mechanism, which can include one or more of object files, software components and deployment descriptors, to facilitate the deployment of the at least one service component.” Although the compiler of ‘492 may be included in at least one embodiment of the instant application as described by the method claims, this does not mean that the compiler described in ‘492 is not patentably distinct from the instant application. As such, the instant application may describe claims that dominate ‘492 without creating a double patenting issue. Thus, Applicants respectfully assert the subject matter of the ‘492 is “patentably distinct” from the instant application.

Clearly, domination and double patenting should not be confused. They are two separate issues. One application may “dominate” a second patent or application when the first application has a broad or generic claim which fully encompasses or reads on an invention defined in a narrower or more specific claim in another patent or application. MPEP § 804 (II.) clarifies that domination by itself, i.e., in the absence of statutory or nonstatutory double patenting grounds, cannot support a double patenting rejection. *In re Kaplan*, 789 F.2d 1574, 1577-78, 229 USPQ 678, 681 (Fed. Cir. 1986); and *In re Sarrett*, 327 F.2d 1005, 1014-15, 140 USPQ 474, 482 (CCPA 1964). In the instant case, some of the claims of the instant application may dominate ones of the copending ‘492 application, but if as in the instant case, the applications disclose and claim “patentably distinct” subject matter, such as overlapping distinct methods and systems, there need not necessarily be a double patenting type rejection made. In fact, the very public policy that creates the judicially created nonstatutory double patenting rejection to prevent an inventor from unjustified or improper timewise extension warrants that patentably distinct items not be refused their own patent. Withdrawal of the provisional rejection is respectfully requested.

Should the Examiner determine that ‘492 and the instant application are “not patentably distinct” as asserted by the Applicants, the Applicants will, upon issuance of either ‘492 or the instant application, submit the necessary Terminal Disclaimer for the remaining application. Thus, there will be no double patenting.

## **CLAIM REJECTIONS UNDER 35 U.S.C.- § 102**

In “Claim Rejections – 35 USC § 102,” item 18 on page 9 of the above-identified final Office Action, claims 1, 4, 10, 11, 16, 17, 22, 38, 39, and 44 have been rejected as being fully anticipated by “Using WebLogic Enterprise JavaBeans” by BEA Systems (hereinafter **BEA WEBLOGIC**) under 35 U.S.C. § 102(b). Applicants respectfully traverse.

As will be explained below, it is believed that the claims were patentable over the cited art in their original form and, therefore, the claims have not been amended to overcome the references.

To anticipate the instant application **BEA WEBLOGIC** must teach every element of the claim as indicated in MPEP 2131, specifically "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). In fact MPEP 2131 clarifies that not only must the claim be expressly or inherently described, but adds that "**The identical invention must be shown** in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989)(emphasis added).

While the Applicants appreciate the demonstration in the above identified office action that the specification of the instant application includes enabling disclosure, the Enterprise JavaBeans, such as an entity bean, described in the specification of the instant application on page 13, do not in and of themselves anticipate the claim language in “as complete detail as is contained in the ... claim” of the instant application. From the specification of the instant application, it is clear that the Enterprise JavaBeans may be automatically generated by the enhanced compiler 106 of the instant application after the compiler is exposed to a stateful annotation such as the “declarative annotations” indicated in claim 1 of the instant application. In contrast, **BEA WEBLOGIC** discloses “encapsulating business logic” in an Enterprise JavaBean, but also requires the developer to “write or obtain an EJBean” and then advises the developer to “pay careful attention” to conform to the different responsibilities of the “EJB specification”, “the EJBean writer”, and “EJB framework”. Moreover, **BEA WEBLOGIC** indicates the need for the developer “to

examine packaged EJBbeans and determine if they follow specified relationships” via a utility called compliancechecker. Thus, under **BEA WEBLOGIC**, **ALL** of the steps must be performed by the **developer**. In contrast, the claimed invention, in claim 1, explicitly relies on “a **compiler** to generate one or more persistent components.” Therefore, the generating of at least some of the persistent components **can not** be performed by a developer. Moreover, **BEA WEBLOGIC** clearly indicates that the business logic is **encapsulated** “inside a component framework” not “**exposed** as part of the stateful web service” as recited in claim 1 of the instant application.

In further contrast to **BEA WEBLOGIC**, the instant application only requires “a source code representation of at least **a portion** of web service logic” as recited in claim 1. Moreover, a compiler of the instant application can use one or more declarative annotations “to generate one or more persistent components to maintain conversational state related to the identified method.”

Moreover, in a specific application of Java to the claims at issue, **BEA WEBLOGIC** does not teach or suggest using an enhanced “compiler to generate” automatically one or more Enterprise JavaBeans™ as well as associated deployment descriptors to store and manage such conversational states based at least in part on “one or more declarative annotations” as recited in claim 1 of the instant application.

Clearly, **BEA WEBLOGIC** does not show, teach or suggest providing “a source code representation of at least a portion of web service logic” and exposing a portion of the logic “as part of the stateful web service” as recited in claim 1 of the instant application.

With respect to Claim 16, in addition to the reasons previously provided, **BEA WEBLOGIC** does not teach or suggest “parsing...source code to identify the presence of one or more declarative annotations” prior to “generating...one or more object codes...based at least in part upon the source code” and “generating ... meta-data based at least in part upon the one or more declarative annotations” as recited in claim 16.

Independent claim 38 contains similar language to claim 16.

It is accordingly believed to be clear that **BEA WEBLOGIC** neither shows nor suggests the features of claims 1, 16, or 38. Claims 1, 16, and 38 are, therefore, believed to be patentable over the art. The dependent claims 4, 10, 11, 17, 22, 39, and 44 are believed to be patentable as well because they all are ultimately dependent on claims 1, 16, or 38.

### **CLAIM REJECTIONS UNDER 35 U.S.C. § 103**

In “Claim Rejections – 35 USC § 103,” item 20 on page 13 of the above-identified final Office Action, claims 2 and 3 have been rejected as being unpatentable over **BEA WEBLOGIC** as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39 and 44 above, and further in view of “EJBDoclet,” December 21, 2000, by dreamBean Software (hereinafter **DREAMBEAN**).

In “Claim Rejections – 35 USC § 103,” item 21 on page 14 of the above-identified final Office Action, claims 5-8, 18, 23-25, 28-30, 40, 45-47, and 50-52 have been rejected as being unpatentable over **BEA WEBLOGIC** as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39 and 44 above, and further in view of “Enterprise JavaBeans” by Monson-Haefel (hereinafter **MONSON-HAEFEL**).

In “Claim Rejections – 35 USC § 103,” item 22 on page 19 of the above-identified final Office Action, claims 9, 19 and 41 have been rejected as being unpatentable over **BEA WEBLOGIC** as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39 and 44 above, and further in view of prior art of record U.S. Patent 5,812,768 to Pagé, et al. (hereinafter **PAGÉ**).

In “Claim Rejections – 35 USC § 103,” item 23 on page 20 of the above-identified final Office Action, claims 12, 31 and 34 have been rejected as being unpatentable over **BEA WEBLOGIC** as applied to claims 1, 4, 10, 11, 16, 17, 22, 38, 39 and 44 above, and further in view of U.S. Patent 6,230,160 to Chan, et al. (hereinafter **CHAN**).

In “Claim Rejections – 35 USC § 103,” item 24 on page 21 of the above-identified final Office Action, claims 13, 20 and 42 have been rejected as being unpatentable over **BEA WEBLOGIC** as applied to claim 1, 4, 10, 11, 16, 17, 22, 38, 39 and 44 above, and further in view of the “Background of the Invention” section appearing on pages 1-3 of the originally filed specification (hereinafter **BOTI**).

In “Claim Rejections – 35 USC § 103,” item 25 on page 22 of the above-identified final Office Action, claim 14 has been rejected as being unpatentable over **BEA WEBLOGIC** and **BOTI** as applied to claims 13, 20 and 42 above, and further in view of **PAGÉ**.

In “Claim Rejections – 35 USC § 103,” item 26 on page 22 of the above-identified final Office Action, claims 15, 21, 26, 27, 43, 48 and 49 have been rejected as being unpatentable over **BEA WEBLOGIC** and **BOTI** as applied to claims 13, 20 and 42 above, and further in view of **MONSON-HAEFEL**.

In “Claim Rejections – 35 USC § 103,” item 27 on page 24 of the above-identified final Office Action, claims 32 and 33 have been rejected as being unpatentable over **BEA WEBLOGIC** and **CHAN** as applied to claims 12, 31, and 34 above, and in further view of **DREAMBEAN**.

In “Claim Rejections – 35 USC § 103,” item 28 on page 24 of the above-identified final Office Action, claim 35 has been rejected as being unpatentable over **BEA WEBLOGIC** and **CHAN** as applied to claim 12, 31 and 34 above, and further in view of **BOTI**.

In “Claim Rejections – 35 USC § 103,” item 29 on page 24 of the above-identified final Office Action, claim 36 has been rejected as being unpatentable over **BEA WEBLOGIC, CHAN** and **BOTI** above, and in further view of **PAGÉ**.

In “Claim Rejections – 35 USC § 103,” item 30 on page 24 of the above-identified final Office Action, claim 37 has been rejected as being unpatentable over **BEA WEBLOGIC, CHAN**, and **BOTI** as applied to claim 36 above, and further in view of **MONSON-HAEFEL**.

All of the rejections provided in “Claim Rejections – 35 USC § 103” of the above identified office action are based in pertinate part on **BEA WEBLOGIC**. As previously discussed, it is believed that the claims were patentable over **BEA WEBLOGIC** in their original form and, therefore, the claims have not been amended to overcome the references.

None of the secondary references, **DREAMBEAN, MONSON-HAEFEL, PAGÉ, CHAN or BOTI** make up for the previously discussed deficiencies of **BEA WEBLOGIC**. Namely, **DREAMBEAN, MONSON-HAEFEL, PAGÉ, CHAN, BOTI**, or acceptable combinations thereof do NOT show providing “a source code representation of at least a portion of web service logic” and exposing a portion of the logic “as part of the stateful web service” as recited in claim 1 of the instant application. Nor do the cited references teach or suggest using “declarative annotations” to cause an enhanced “compiler to generate ... persistent components” as recited in claim 1 of the instant application.

Furthermore, **DREAMBEAN, MONSON-HAEFEL, PAGÉ, CHAN, BOTI**, or acceptable combinations thereof do not teach or suggest “parsing...source code to identify the presence of one or more declarative annotations” prior to “generating...one or more object codes...based at least in part upon the source code” and “generating ... meta-data based at least in part upon the one or more declarative annotations” as recited in claim 16

Independent claims 23, 31, 38, and 45 contain similar language to claims 1 and 16.

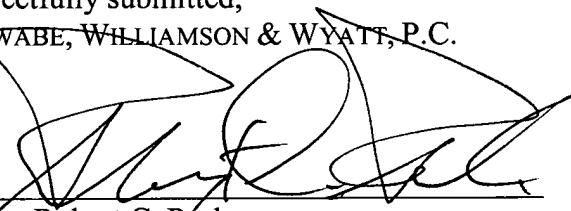
It is accordingly believed to be clear that none of the references, whether taken alone or in any combination, either show or suggest the features of claims 1, 16, 23, 31, 38, or 45. The independent claims 1, 16, 23, 31, 38, and 45 are, therefore, believed to be patentable over the art. The dependent claims are believed to be patentable as well because they all are ultimately dependent on claims 1, 16, 23, 31, 38, or 45.

If an extension of time is required, petition for extension is herewith made. Any extension fee associated therewith should be charged to the Deposit Account of Schwabe, Williamson and Wyatt, P.C., No. 50-0393.

## CONCLUSION

In view of the foregoing, reconsideration and allowance of claims 1-52 are solicited. As a result of the amendments made herein, Applicant submits that claims 1-52 are in condition for allowance. Accordingly, a Notice of Allowance if respectfully requested. If the Examiner has any questions concerning the present paper, the Examiner is kindly requested to contact the undersigned at (206) 407-1509. If any fees are due in connection with filing this paper, the Commissioner is authorized to charge the Deposit Account of Schwabe, Williamson and Wyatt, P.C., No. 50-0393.

Respectfully submitted,  
SCHWABE, WILLIAMSON & WYATT, P.C.

by:   
Robert C. Peck  
Reg. No.: 56,826

Date: October 3, 2005

Schwabe, Williamson & Wyatt, P.C.  
Pacwest Center, Suites 1600-1900  
1211 SW Fifth Avenue  
Portland, Oregon 97222  
Telephone: 503-222-9981